Вестник Донского государственного технического университета. 2019. Т. 19, № 2. С. 185-194. ISSN 1992-5980 eISSN 1992-6006
Vestnik of Don State Technical University. 2019. Vol. 19, no. 2, pp. 185-194. ISSN 1992-5980 eISSN 1992-6006



УДК 004.942 https://doi.org/10.23947/1992-5980-2019-19-2-185-194

Сравнительный анализ модифицированной постквантовой криптографической системы NTRUENcrypt и общепринятой криптосистемы RSA*

П. В. Разумов[1], И. А. Смирнов[2], И. А. Пилипенко[3], А. В. Селёва[4], Л. В. Черкесова[5]**
[1,2,3,4,5]Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

Comparative analysis of NTRUEncrypt modified post-quantum cryptographic system and standard RSA cryptosystem***

P. V. Razumov[1], I. A. Smirnov[2], I. A. Pilipenko[3], A. V. Selyova[4], L. V. Cherkesova[5]**
[1,2,3,4,5] Don State Technical University, Rostov-on-Don, Russian Federation

Введение. Статья посвящена исследованию криптографической системы NTRUEncrypt, расчету алгоритмической сложности разработки криптосистемы NTRUEncrypt и ее модификации. Цели исследования: разработка эффективного постквантового криптографического алгоритма NTRUEncrypt, обладающего высокой криптостойкостью к атакам с квантового компьютера, а также разработка модификации предложенного алгоритма, анализ и экспериментальное доказательство его преимуществ.
Материалы и методы. Предложено описание системы шифрования NTRUEncrypt. Изучена модификация рассматриваемого алгоритма, представлена блок-схема реализации основанного на нем программного средства. Приведен пример работы программного средства и дана его характеристика. Достоверность результатов обоснована с помощью *U*-критерия Манна — Уитни. При проведении эксперимента использована сторонняя программная реализация криптографической системы RSA. В исходный код всех трех программ NTRUEncrypt, RSA, модификации NTRUEncrypt был внедрен элемент класса Stopwatch. Данный класс предоставляет набор методов и свойств, которые можно использовать для точного измерения времени, затраченного на выполнение. Таким образом, появилась возможность фиксировать результаты затраченного времени на всех трех основных этапах: создание ключей, шифрование и расшифрование сообщения.
Результаты исследования. Доказаны преимущества разработанных криптосистем по характеристикам производительности. Выполнено экспериментальное сравнение реализованного алгоритма NTRUEncrypt и его модификации. При этом обозначены все преимущества последней.
Обсуждение и заключения. Экспериментально доказано преимущество использования модификации алгоритма NTRUEncrypt. Новое приложение на 25 % быстрее выполняет общую работу по генерации ключей, шифрованию и расшифрованию. Помимо этого оптимизируется использование внутренней памяти за счет уменьшения веса ис-

Introduction. The NTRUEncrypt cryptographic system, the calculation of the algorithmic complexity of the development of the NTRUEncrypt cryptosystem and its modifications are considered. The study objectives are to develop NTRUEncrypt, an efficient post-quantum cryptographic algorithm, which has high cryptographic resistance to quantum computer attacks, to work out a modification of the proposed algorithm, to analyze and experimentally validate its advantages.
Materials and Methods. A description of the NTRUEncrypt encryption system is proposed. The modification of the considered algorithm is studied; the block diagram of the implementation of the software based on it is presented. An example of the software operation and its characteristic is given. The reliability of the results is proved using the Mann-Whitney U test. During the experiment, the third-party software implementation of the RSA cryptosystem was used. A Stopwatch class element was introduced in the source code of all three programs of NTRUEncrypt, RSA, and NTRUEncrypt modifications. This class provides a set of methods and properties that can be used for the precise measurement of the execution time. Thus, it became possible to record the results of the time spent on all three basic stages: key creation, encryption and decryption of the message.
Research Results. The advantages of the developed cryptosystems in terms of the performance characteristics are proved. An experimental comparison of the implemented NTRUEncrypt algorithm and its modification is performed. All advantages of the latter are indicated.
Discussion and Conclusions. The advantage of using the NTRUEncrypt algorithm modification is experimentally validated. The new application is 25% faster to perform general work on key generation, encryption and decryption. In addition, the internal memory usage is optimized through reducing the weight of the source program file and the size of the secret

*Работа выполнена в рамках инициативной НИР.
**E-mail: therazumov@gmail.com, terran.doatk@mail.ru, ipilipenko@donstu.ru, tone4ka.selyova@yandex.ru, chia2002@inbox.ru
***The research is done within the frame of the independent R&D.

ходного файла программы и размера секретного ключа. При попытке взлома шифротекста проявляется криптографическая стойкость и сложность использования квантовых алгоритмов.

key. When attempting to crack a ciphertext, cryptographic robustness and complexity of using quantum algorithms are shown.

Ключевые слова: криптографическая система, постквантовый криптографический алгоритм, криптостойкость, U-критерий Манна — Уитни, шифрование.

Keywords: cryptographic system, post-quantum cryptographic algorithm, cryptographic strength, Mann-Whitney U test, encryption.

Образец для цитирования: Сравнительный анализ модифицированной постквантовой криптографической системы NTRUENcrypt и общепринятой криптосистемы RSA / П. В. Разумов [и др.] // Вестник Дон. гос. техн. ун-та. — 2019. — Т. 19, № 2. — С.185−194. https://doi.org/10.23947/1992-5980-2019-19-2-185-194

For citation: P.V. Razumov, et al. Comparative analysis of NTRUEncrypt modified post-quantum cryptographic system and standard RSA cryptosystem. Vestnik of DSTU, 2019, vol. 19, no. 2, pp. 185−194. https://doi.org/10.23947/1992-5980-2019-19-2-185-194

Введение. Толчком для разработки новой криптографической системы послужила статья [1]. В ней показано, что квантовые компьютеры потенциально угрожают взломом всем широко используемым криптографическим алгоритмам.

Представленная в данной статье программная разработка обладает криптостойкостью по отношению к возможным квантовым атакам и превосходит аналоги (например, криптосистему RSA) по характеристикам скорости работы алгоритма и по количеству затрачиваемых ресурсов [2]. Этим обусловлена актуальность работы.

Объектом исследования является криптосистема NTRUEncrypt.

Предмет исследования — алгоритмическая сложность разработки криптосистемы NTRUEncrypt и ее модификации.

Цели исследования: разработка эффективного постквантового криптографического алгоритма NTRUEncrypt, обладающего высокой криптоскойстью по отношению к атакам с квантового компьютера, а также создание модификации предложенного алгоритма, анализ и экспериментальное доказательство его преимуществ.

В соответствии с поставленной целью были определены следующие задачи.

1. Исследовать алгоритм работы криптосистемы NTRUEncrypt.
2. Разработать алгоритм модификации NTRUEncrypt.
3. Реализовать программные средства криптосистемы NTRUEncrypt и ее модификации.
4. Проанализировать и сравнить две программы друг с другом и с их аналогом — криптосистемой RSA.

Материалы и методы. Рассмотрим описание системы шифрования NTRUEncrypt. Криптографическая система с открытым ключом NTRUEncrypt использует операции над кольцом $Z\,[X]\,/\,(X^N-1)$ многочленов степени, не превосходящей $N-1$ [3]:

$$a = a_0 + a_1 * X^1 + a_0 * X^2 + \cdots + a_{N-1} * X^{N-1},$$

где $a_0, a_1, a_2 \ldots\ a_{N-1}$ — целые числа.

Операции сложения и умножения производятся как обычно, за исключением того, что X^N заменяется на 1, X^{N+1} заменяется на X^1, X^{N+2} заменяется на X^2 и т. д.

Криптосистема определяется рядом параметров, основные из которых: N, p и q. Для сохранения стойкости алгоритма необходимо, чтобы параметры p и q были взаимно простыми.

Чтобы обеспечить высокую стойкость алгоритма к различным атакам, рекомендуется использовать следующие параметры (рис. 1):

Обозначение	N	q	p	df	dg	dr	*Гарантированная стойкость*
NTRU167:3	167	128	3	61	20	18	Умеренный уровень стойкости
NTRU251:3	251	128	3	50	24	16	Стандартный уровень стойкости
NTRU503:3	503	256	3	216	72	55	Высочайший уровень стойкости
NTRU167:2	167	127	2	45	35	18	Умеренный уровень стойкости
NTRU251:2	251	127	2	35	35	22	Стандартный уровень стойкости
NTRU503:2	503	253	2	155	100	65	Высочайший уровень стойкости

Рис. 1. Рекомендованные параметры

Результаты исследования

Генерация ключей. Боб хочет передать сообщение Алисе. Для этого ему необходимы открытый и закрытый ключи. Поэтому он случайным образом выбирает два малых полинома f и g из кольца усеченных многочленов R. Малость полиномов означает, что относительно произвольного полинома по модулю q, в котором коэффициенты равномерно распределены, у малого полинома они будут много меньше q [4]. Для определения малости полиномов используются числа df и dg, которые Боб выбирает самостоятельно.

Полином f будет иметь df коэффициентов, равных единице, ($df - 1$) коэффициентов, равных минус единице, и остальные, равные нулю.

Полином g будет иметь dg коэффициентов, равных единице, столько же коэффициентов, равных минус единице, и остальные, равные нулю.

Боб должен хранить выбранные полиномы в тайне, так как любой, кому они станут известны, сможет расшифровать сообщение.

Следующим шагом Боб вычисляет обратные полиномы f_p и f_q по модулю p и q соответственно, такие, что:

$$f \times f_p = 1(mod\ p) \text{ и } f \times f_q = 1(mod\ q).$$

Если случайно эти обратные полиномы не существуют, то Боб возвращается назад и заново выбирает полином f.

Секретный ключ — это пара (f, f_p), а открытый ключ h вычисляется по формуле:

$$h = p \times f_q \times g\ (mod\ q).$$

Шифрование. Алиса хочет отправить сообщение Бобу с помощью открытого ключа h. Для этого Алисе нужно представить свое сообщение в виде полинома m с коэффициентами по модулю p, выбранными из диапазона $(-p/2, p/2]$. Затем Алисе необходимо выбрать другой малый полином r, который называется «ослепляющий», и вычислить шифротекст:

$$e = (r \times h + m)(mod\ q).$$

Расшифрование. Боб получает от Алисы зашифрованное сообщение e и хочет его расшифровать. Первым делом, используя свой секретный ключ, Боб вычисляет:

$$a = f \times e\ (mod\ q).$$

Так как Боб вычисляет значение a по модулю числа q, он должен выбрать его коэффициенты из диапазона $(-q/2, q/2]$ и затем вычислить:

$$b = a\ (mod\ p).$$

Наконец, Боб, используя вторую часть секретного ключа, получает исходное сообщение от Алисы:

$$c = f_p \times b\ (mod\ p).$$

Модификация алгоритма NTRUEncrypt. Как видно из описания алгоритма, полином f должен отвечать следующим требованиям:

— полином f является обратимым по модулю p,

— полином f является обратимым по модулю q,

— полином f является малым полиномом.

В самом алгоритме обратимость по модулю p и q гарантировалась следующим образом. Если вдруг генерировался полином f, не обратимый по одному из модулей, то он отбрасывался и генерировался следующий — и так до тех пор, пока необходимый полином не будет найден.

Предлагаемая модификация заключается в том, чтобы заменить полином f полиномом вида:

$$f = 1 + pF, \tag{1}$$

где F — малый полином.

Данный подход имеет следующие преимущества.

1. Из самого выражения (1) видно, что полином f всегда обратим по модулю p. Этот факт ускоряет генерацию ключей, так как не требуется дополнительно вычислять f_p.

2. Поскольку $f^{-1} = 1\ mod\ p$, то при расшифровании не требуется дополнительное умножение на f^{-1}, что ускоряет сам процесс расшифровки. Закрытый ключ в этом случае будет представлять собой не пару (f, f_p), а (f).

Генерация ключей. Как и в оригинальном алгоритме, вначале Боб выбирает параметры шифрования N, p, q и числа df, dg. Затем он случайным образом выбирает два малых полинома F и g из кольца усеченных многочленов R.

Вычисляет модифицированный полином f по формуле (1).

Следующим шагом Боб вычисляет обратный полином f_q по модулю q:

$$f \times f_q = 1(mod\ q).$$

Если случайно обратный полином не находится, Боб возвращается назад и заново выбирает полином f.

Секретный ключ — это полином f, а открытый ключ h вычисляется следующим образом:

$$h = p \times f_q \times g\ (mod\ q).$$

Шифрование. Шифрование остается без изменений, все в точности как в оригинальном алгоритме NTRUEncrypt.

Алиса хочет отправить сообщение Бобу с помощью открытого ключа h. Для этого Алисе нужно представить свое сообщение в виде полинома m с коэффициентами по модулю p, выбранными из диапазона $(-p/2, p/2]$. Затем Алисе необходимо выбрать другой малый полином r, который называется «ослепляющий», и вычислить шифротекст:

$$e\ =\ (r \times h + m)(mod\ q).$$

Расшифрование. Боб получает от Алисы зашифрованное сообщение e и хочет его расшифровать. Первым делом, используя свой секретный ключ, Боб вычисляет:

$$a = f \times e\ (mod\ q).$$

Так как Боб вычисляет значение a по модулю числа q, он должен выбрать его коэффициенты из диапазона $(-q/2, q/2]$ и затем вычислить:

$$b = a\ (mod\ p).$$

Всё, на этом вычисления закончены, мы получили исходное сообщение от Алисы: $b = m$ [5].

Доказательство работы модифицированного алгоритма. Для доказательства работы алгоритма рассмотрим сам процесс расшифрования.

Зашифрованное сообщение от Алисы имеет вид:

$$e\ =\ (r \times h + m)(mod\ q).$$

Боб использует свой закрытый ключ — полином f:

$$a = f \times e\ (mod\ q)\ = (f\ \times\ (r\ \times\ h\ +\ m))(mod\ q)\ = (f\ \times\ (r\ \times\ pf_q \times g +\ + m))(mod\ q).$$

В результате:

$$a = (pr\ \times\ g\ +\ m\ \times\ f)(mod\ q).$$

Следующим шагом Боб получает полином b путем уменьшения коэффициентов полинома a по модулю p:

$$b = a(modp) = m \times f(modp) = (m\ +\ m\ \times\ p\ \times\ F)(mod\ p) = m\ (mod\ p).$$

Таким образом, мы проверили и доказали, что полином b действительно является исходным сообщением m.

Реализация алгоритма. В качестве языка программирования использован объектно-ориентированный язык программирования *C#*, относящийся к семье языков с *C*-подобным синтаксисом. Среда разработки — Microsoft Visual Studio 2015 Enterprise. Главное преимущество данного программного продукта — приложение с графическим интерфейсом, который позволяет пользователю быстро разобраться в устройстве и схеме работы данного программного продукта.

На рис. 2 приведена обобщенная блок-схема работы программного средства.

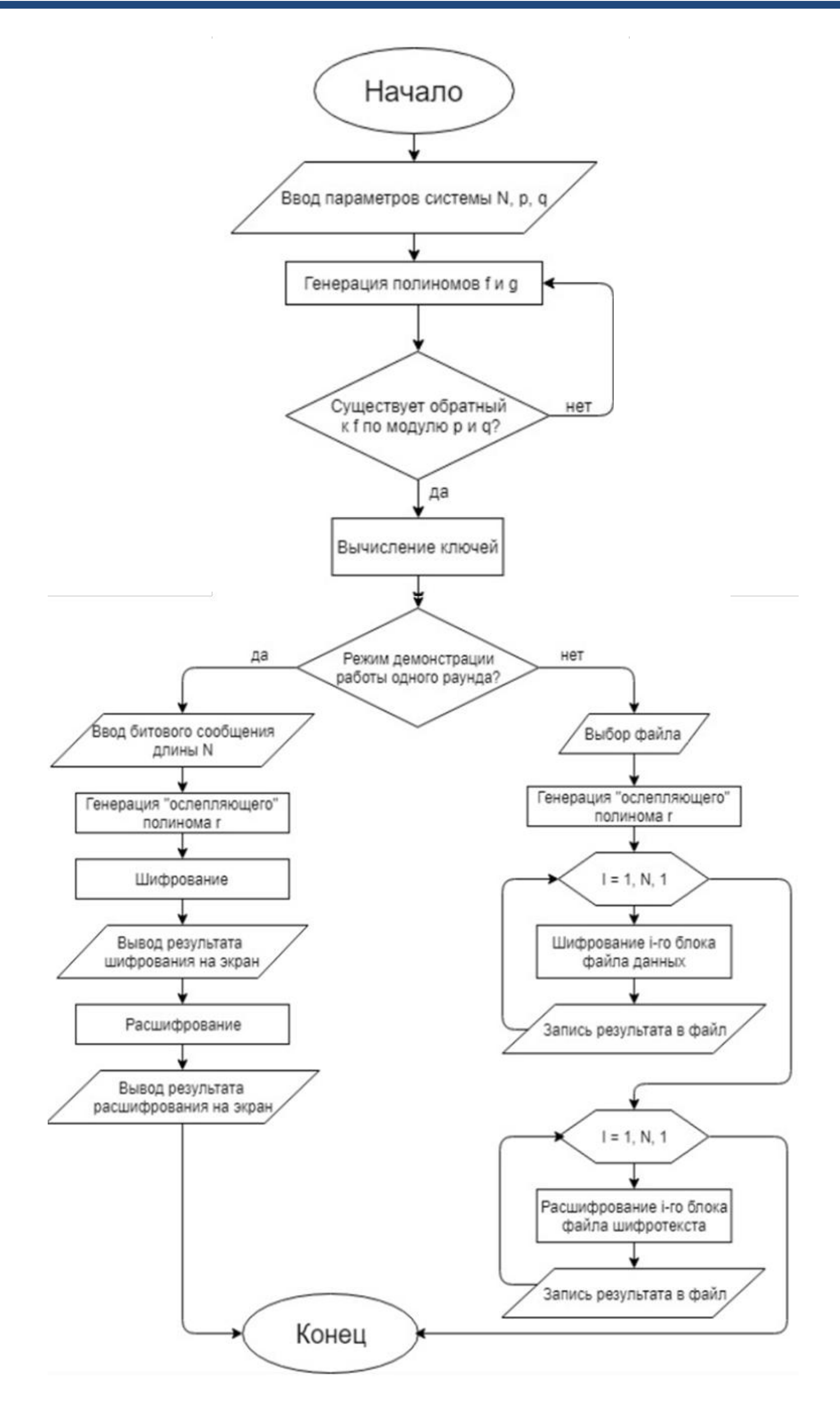


Рис. 2. Блок-схема алгоритма реализации программного средства

Пример работы программного средства. NTRUEncrypt использует три постоянных параметра: N, p, q. Они вводятся пользователем на панели «Параметры системы».

Далее располагается панель «Генерация ключей». На вход подаются параметры, введенные пользователем на предыдущем этапе. Затем случайным образом формируются два полинома f и g — такие, чтобы:

— количество коэффициентов, равных единице и нулю, было равно числу, которое заранее определено в программе;

— степень полиномов соответствовала введенному параметру N.

Затем используется теорема Евклида о нахождении наибольшего общего делителя для многочленов и ее обратный ход для вычисления двух обратных к f полиномов по модулю p и q соответственно. Вычисляется открытый ключ h.

Пользователь вводит двоичный вид сообщения m на панели «Шифрование» в окне «Сообщение». Стоит отметить, что исходное сообщение необходимо разбить на блоки по N бит, каждый из которых будет обрабатываться отдельно и преобразовывать каждый блок в полином с коэффициентами {–1, 0, 1} (в данной программе вместо значения «–1» используется «2».

Для операции шифрования программе необходимо выполнить предварительную подготовку — сгенерировать один ослепляющий полином. Он формируется по такому же принципу, что и полиномы f, g.

Полученный открытый ключ и полином r дают возможность зашифровать сообщение m, используя соответствующую формулу. Затем результат проверяется на принадлежность кольцу усеченных многочленов степени, не превосходящей $N - 1$, и выводится на экран.

Следующий блок реализует механизм дешифрации. Для этого последовательно выполняются действия с учетом кольца $Z[X]/(X^N - 1)$.

Результат проведенных операций отображен на рис. 3.

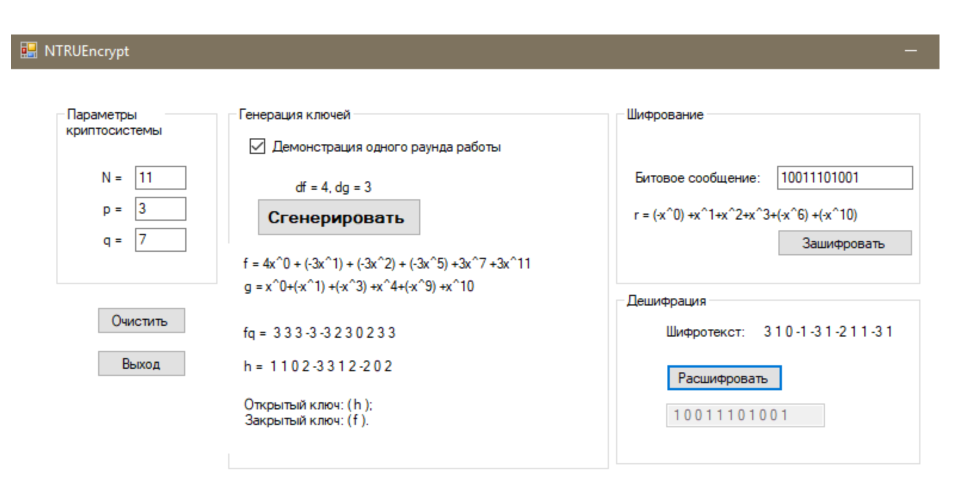


Рис. 3. Внешний вид программы NTRUEncrypt

На рис. 4 отображены результаты работы программного средства в виде исходного текста, зашифрованного и расшифрованного.

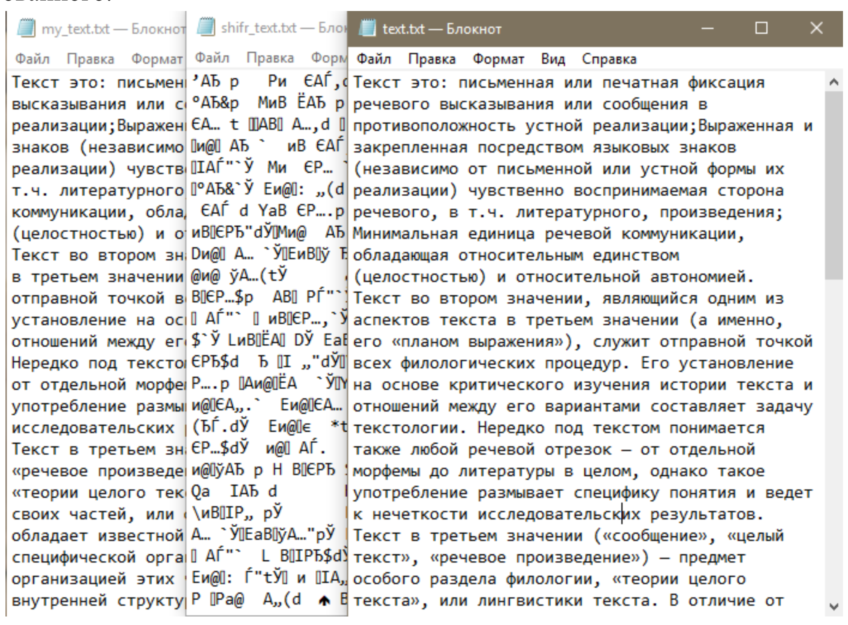


Рис. 4. Результат работы программного средства

Эксперименты. Цель экспериментальное исследования — показать и доказать преимущества по характеристикам производительности разработанных криптосистем в сравнении с аналогом. (В рамках данной работы в качестве альтернативной была выбрана криптосистема с открытым ключом RSA [6].) Кроме того, необходимо было сравнить реализованный алгоритм NTRUEncrypt и его модификацию, чтобы отметить преимущества последней и привести экспериментальные доказательства.

Для обоснования достоверности результатов использовался статистический U-критерий Манна — Уитни, который используется для сравнения двух независимых выборок по уровню какого-либо признака, изме-

ренного количественно. Критерий позволяет определить степень различия между выборками и является более мощным по сравнению с критерием Розенбаума.

Для проведения эксперимента была использована сторонняя программная реализация криптографической системы RSA. В исходный код всех трех программ NTRUEncrypt, RSA, модификации NTRUEncrypt был внедрен элемент класса Stopwatch. Данный класс предоставляет набор методов и свойств, которые можно использовать для точного измерения времени, затраченного на выполнение. Таким образом, появилась возможность фиксирования результатов затраченного времени на всех трех основных этапах: создание ключей, шифрование и расшифрование сообщения. Результаты эксперимента представлены в табл. 1.

Таблица 1

Результаты эксперимента

№	Время работы NTRUEncrypt, сек	Время работы модификации NTRUEncrypt, сек	Время работы RSA, сек
Генерация ключей			
1	0,0239676	0,0190014	0,0964144
2	0,0149743	0,0109994	0,1023557
3	0,0129915	0,0099936	0,1372658
4	0,0170503	0,0099772	0,0491137
5	0,0139885	0,0099773	0,0869555
6	0,0139880	0,0099782	0,0587503
7	0,0129761	0,0099936	0,0986608
8	0,0139931	0,0109775	0,0707417
9	0,0139918	0,0099773	0,0595015
10	0,0169748	0,0099777	0,0517874
Шифрование			
1	0,0069961	0,0069770	0,0453772
2	0,0049961	0,0059796	0,0598658
3	0,0059954	0,0049965	0,0265461
4	0,0049966	0,0069966	0,0402407
5	0,0069765	0,0060854	0,0783415
6	0,0049961	0,0069989	0,0097463
7	0,0059795	0,0059954	0,0247876
8	0,0069961	0,0059954	0,0272320
9	0,0089956	0,0049798	0,0494907
10	0,0059954	0,0049744	0,0100916
Расшифрование			
1	0,0009843	0,0009843	1,0624476
2	0,0010212	0,0010026	0,3692022
3	0,0010193	0,0009974	0,7651423
4	0,0009989	0,0009975	0,9922103
5	0,0030012	0,0009988	0,3056423
6	0,0010017	0,0009975	0,3757183
7	0,0009989	0,0009988	1,2844272
8	0,0010021	0,0009989	0,5902735
9	0,0010026	0,0009984	0,9932694
10	0,0009844	0,0009970	0,8740392

К результатам, приведенным в табл. 1, применим *U*-критерий Манна —Уитни. Сравним скорость генерации ключей алгоритма NTRUEncrypt и его модификации.

Следуя алгоритму *U*-критерия Манна — Уитни, распишем вычисление критерия по шагам.

1. Составить общий ранжированный список из обеих выборок, присвоив меньшему рангу меньшее значение.

2. Разделить общий ранжированный список на два, состоящих из элементов первой и второй выборок.

3. Подсчитать сумму рангов, приходящихся на первую и вторую выборки по отдельности, как показано на рис. 5.

№	Выборка 1	Ранг 1	Выборка 2	Ранг 2
1	0.0239676	20	0.0190014	19
2	0.0149743	16	0.0109994	9
3	0.0129915	11	0.0099936	6.5
4	0.0170503	18	0.0099772	1
5	0.0139885	13	0.0099773	2.5
6	0.0139880	12	0.0099782	5
7	0.0129761	10	0.0099936	6.5
8	0.0139931	15	0.0109775	8
9	0.0139918	14	0.0099773	2.5
10	0.0169748	17	0.0099777	4
Суммы:		146		64

Рис. 5. Второй шаг вычисления *U*-критерия Манна — Уитни

4. Вычислить значение *U*-критерия Манна — Уитни: $U = 9$. По таблице для уровня статистической значимости следует определить критическое значение критерия для данных $n1$ и $n2$ (рис. 6).

$U_{Кр}$	
p≤0.01	p≤0.05
19	**27**

Рис. 6. Четвертый шаг вычисления *U*-критерия Манна — Уитни

Так как значения $n1$ и $n2$ для всех экспериментов одинаковы, то эта таблица будет использоваться в каждом вычислении.

Отсюда можно сделать вывод, что полученное эмпирическое значение $U = 9$ находится в зоне значимости (рис. 7). Следовательно, наблюдается значительное различие между скоростью работы программы NTRUEncrypt и ее модификацией. Если этот факт выразить в процентном отношении, получится, что модификация NTRUEncrypt быстрее самой программы на 28 %.

Ось значимости:

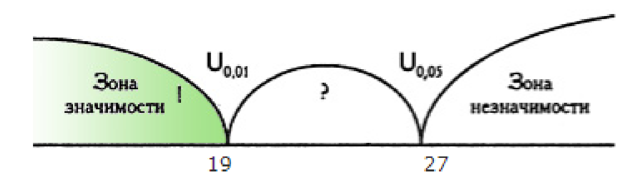


Рис. 7. Ось значимости

Аналогично, опуская подробности вычислений, приведем результаты *U*-критерия Манна — Уитни и процентное превосходство для остальных случаев.

Генерация ключей

1. Модификация NTRUEncrypt быстрее программы NTRUEncrypt на 28 %. Полученное эмпирическое значение $U = 9$ находится в зоне значимости.

2. Программа NTRUEncrypt быстрее RSA на 80 %. Полученное эмпирическое значение $U = 0$ находится в зоне значимости.

3. Модификация NTRUEncrypt быстрее RSA на 86 %. Полученное эмпирическое значение $U = 0$ находится в зоне значимости.

Шифрование сообщения

1. Полученное эмпирическое значение $U = 47$ находится в зоне незначимости.

2. Программа NTRUEncrypt быстрее RSA на 80 %. Полученное эмпирическое значение $U = 0$ находится в зоне значимости.

3. Модификация NTRUEncrypt быстрее RSA на 84 %. Полученное эмпирическое значение $U = 0$ находится в зоне значимости.

Расшифрование

1. Модификация NTRUEncrypt быстрее программы NTRUEncrypt на 17 %. Полученное эмпирическое значение $U = 24$ находится в зоне неопределенности.

2. Программа NTRUEncrypt быстрее RSA на 99 %. Полученное эмпирическое значение $U = 0$ находится в зоне значимости.

3. Модификация NTRUEncrypt быстрее RSA на 99 %. Полученное эмпирическое значение $U = 0$ находится в зоне значимости.

Обсуждение и заключения. В рамках данной работы созданы:
— программное обеспечение, реализующее работу криптосистемы NTRUEncrypt;
— программное средство, реализующее модификацию этой криптосистемы.

Из преимуществ NTRUEncrypt перед аналогом — криптосистемой RSA можно указать более высокую скорость работы. Выполнение операций шифрования и расшифрования требует $O(n^2)$ операций, в отличие от $O(n^3)$ у того же RSA. По экспериментальным данным программа NTRUEncrypt значительно выигрывает по скорости работы алгоритма в сравнении с RSA. Кроме того, отмечается небольшое увеличение стойкости при фактически такой же длине ключа. Недостаток системы — необходимость использования рекомендованных параметров.

Что касается стойкости NTRUEncrypt, то после создания квантовых компьютеров будут решены задачи быстрой факторизации и дискретного логарифмирования [7]. В этом случае RSA, DSA и подобные им алгоритмы станут бесполезными. Актуальность NTRUEncrypt сохранится: он будет вполне применим и в «постквантовую» эпоху, т. к. не существует алгоритма, решающего задачу кратчайшего вектора решетки.

Экспериментально доказано преимущество использования модификации алгоритма NTRUEncrypt. Разработанное приложение на 25 % быстрее выполняет общую работу по генерации ключей, шифрованию и расшифрованию. Кроме того, оптимизируется использование внутренней памяти за счет уменьшения веса исходного файла программы и размера секретного ключа. При попытке взлома шифротекста проявляется криптографическая стойкость и сложность использования квантовых алгоритмов.

Библиографический список

1. Shor, P. Algorithms for Quantum Computation: Discrete Log and Factoring / P. Shor. — Murray Hill : AT&T Bell Labs, 1994. — 124–134 p.

2. Шаклеина, Т. А. «Мозговые центры» и их роль в формировании внешней политики США / Т. А. Шаклеина // Введение в прикладной анализ международных ситуаций. — Москва : Аспект Пресс, 2014. — С. 112.

3. Основы криптографии / А. П. Алферов [и др.]. — Москва : Гелиос АРВ, 2002. — С. 209–220.

4. Лапонина, О. Р. Криптографические основы безопасности / О. Р. Лапонина. — Москва: Национальный открытый университет ИНТУИТ, 2016. — С. 118.

6. Ишмухаметов, Ш. Т. Методы факторизации натуральных чисел / Ш. Т. Ишмухаметов. — Казань : Изд-во Казан. ун-та, 2011. — С. 74–82.

5. Bakhtiari, M. Serious Security Weakness in RSA Cryptosystem / M. Bakhtiari, M. A. Maarof // International Journal of Computer Science and Information Security. — 2012. — № 3. — P. 175–178.

7. Василенко, О. Н. Теоретико-числовые алгоритмы в криптографии / О. Н. Василенко. — Москва : МЦНМО, 2003. — С. 73–74.

Сдана в редакцию 23.10.2018
Принята к публикации 14.03.2019

Об авторах:

Разумов Павел Владимирович,
Студент кафедры «Кибербезопасность информационных систем» Донского государственного технического университета, (РФ, 344000, г. Ростов-на-Дону, пл. Гагарина, 1),
ORCID: https://orcid.org/0000-0003-2454-3600
therazumov@gmail.com

Смирнов Иван Андреевич,
студент кафедры «Кибербезопасность информационных систем» Донского государственного технического университета, (РФ, 344000, г. Ростов-на-Дону, пл. Гагарина, 1),
ORCID: https://orcid.org/0000-0001-6533-4368
terran.doatk@mail.ru

Пилипенко Ирина Александровна,
аспирант кафедры «Кибербезопасность информационных систем» Донского государственного технического университета, (РФ, 344000, г. Ростов-на-Дону, пл. Гагарина, 1),
ORCID: https://orcid.org/0000-0003-3236-6069
ipilipenko@donstu.ru

Селёва Антонина Владимировна,
студент кафедры «Кибербезопасность информационных систем» Донского государственного технического университета, (РФ, 344000, г. Ростов-на-Дону, пл. Гагарина, 1),
ORCID: https://orcid.org/0000-0003-0990-7429
tone4ka.selyova@yandex.ru

Черкесова Лариса Владимировна,
профессор кафедры «Кибербезопасность информационных систем» Донского государственного технического университета, (РФ, 344000, г. Ростов-на-Дону, пл. Гагарина, 1), доктор технических наук, профессор,
ORCID: https://orcid.org/0000-0002-9392-3140
chia2002@inbox.ru